“© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Hyper-Parameter Initialization for Squared Exponential Kernel-based Gaussian Process Regression

Nalika Ulapane
Electrical and Electronic Engineering
The University of Melbourne
Parkville VIC 3010, Australia
nalika.ulapane@unimelb.edu.au

Karthick Thiyagarajan
UTS Robotics Institute
University of Technology Sydney
Ultimo NSW 2007, Australia
karthick.thiyagarajan@uts.edu.au

Sarath Kodagoda
UTS Robotics Institute
University of Technology Sydney
Ultimo NSW 2007, Australia
sarath.kodagoda@uts.edu.au

Abstract—**Hyper-parameter optimization is an essential task in the use of machine learning techniques. Such optimizations are typically done starting with an initial guess provided to hyper-parameter values followed by optimization (or minimization) of some cost function via gradient-based methods. The initial values become crucial since there is every chance for reaching local minimums in the cost functions being minimized, especially since gradient-based optimizing is done. Therefore, initializing hyper-parameters several times and repeating optimization to achieve the best solutions is usually attempted. Repetition of optimization can be computationally expensive when using techniques like Gaussian Process (GP) which has an $\mathcal{O}(n^3)$ complexity, and not having a formal strategy to initialize hyper-parameter values is an additional challenge. In general, re-initialization of hyper-parameter values in the contexts of many machine learning techniques including GP has been done at random over the years; some recent developments have proposed some initialization strategies based on the optimization of some meta loss cost functions. To simplify this challenge of hyper-parameter initialization, this paper introduces a data-dependent deterministic initialization technique. The specific case of the squared exponential kernel-based GP regression problem is focused on, and the proposed technique brings novelty by being deterministic as opposed to random initialization, and fast (due to the deterministic nature) as opposed to optimizing some form of meta cost function as done in some previous works. Although global suitability of this initialization technique is not proven in this paper, as a preliminary study the technique's effectiveness is demonstrated via several synthetic as well as real data-based nonlinear regression examples, hinting that the technique may have the effectiveness for broader usage.**

Index Terms—**Gaussian Process, hyper-parameters, kernel, machine learning, nonlinear regression, optimization, squared exponential.**

I. Introduction

Gaussian Processes (GP)s are a strong machine learning tool used for solving multi-dimensional nonlinear regression problems as well as classification problems [1], [2]. It is used for numerous applications varying from modeling and calibration to forecasting and predictive control-related applications [3]–[8]. GP is considered a non-parametric approach, and works by means of finding some optimized hyper-parameter values. Learning suitable hyper-parameter values for GP is typically done via gradient-based optimization (targeted at minimizing the negative log marginal likelihood, described in Section II) which utilizes training data and user-specified initial values given for hyper-parameters, and GP learning is known to be $\mathcal{O}(n^3)$ complex. Usually, optimizing starting from multiple initialization points can be required to avoid reaching local minimums of the cost function and achieve the best solutions. GP related hyper-parameter initializing as such has conventionally been done at random [2], and this paper contributes to this space by introducing a more formal data-dependent deterministic initialization technique.

Speeding up GP due to its $\mathcal{O}(n^3)$ complexity has been of interest over the years, and sparse GP approaches that use a subset of training data to estimate hyper-parameters have been proposed and studied as a solution [9], [10]. Hyper-parameter optimization algorithms have been studied separately to be general for many machine learning techniques [11]–[14]; however, strategies for initializing hyper-parameters have not been a primary focus. The work on initializing hyper-parameters has mainly revolved around some form of meta learning [15], [16]. In contrast to such previous works, this paper contributes by introducing a hyper-parameter initialization technique that is data-dependent and deterministic in relation to GP-based regression. The specific case of using the squared exponential kernel is considered as a preliminary study, focusing on producing nonlinear regression models for processes exhibiting continuous and smooth functional behavior. The initialization technique relies on the absolute difference between adjacent training data points. This paper presents the mathematical formulation of the initialization technique along with some demonstrative examples that show the technique's effectiveness in producing regression models having low residual errors and reasonable uncertainty.

The structure of the paper is as follows: Section II describes the principles of GP-based regression; Section III proposes the hyper-parameter initialization technique; Section IV presents the effectiveness of the proposed technique via some synthetic and real data-based regression examples, and Section V concludes the paper by discussing the implications of results and potential avenues for future work.

II. Related Knowledge: Gaussian Process Formulation for Regression

Reference [1] is useful to study GPs in detail while this section summarizes the GP regression formulation for nonlinear regression. Works [17]–[25] provide some examples for the use of GP for solving nonlinear regression problems.

Suppose $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^{m}$, $i, m \in \mathbb{Z}^{+}$ is a training set of independent identically distributed (i.i.d.) examples having some unknown distribution, drawn from the noisy process:

$$y^{(i)} = f(x^{(i)}) + \epsilon^{(i)}, \tag{1}$$

$i = 1, 2, ..., m$, where $\epsilon^{(i)}$ are i.i.d. 'noise' variables with independent $\mathcal{N}(0, \sigma^2)$ distributions. A 'prior distribution' over functions $f(\cdot)$ is assumed; in particular, a zero-mean GP prior,

$$f(\cdot) \sim \mathcal{GP}(0, k(\cdot, \cdot)) \tag{2}$$

is assumed, for some valid covariance function $k(\cdot, \cdot)$. Now, suppose $T = \{(x_*^{(i)}, y_*^{(i)})\}_{i=1}^{m_*}$ is a set of i.i.d. testing points drawn from the same unknown distribution as S. For notational simplicity, the following are defined.

$$X = \begin{bmatrix} -(x^{(1)})^T- \\ -(x^{(2)})^T- \\ \vdots \\ -(x^{(m)})^T- \end{bmatrix} \in \mathbb{R}^{m \times n}, \ \overrightarrow{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \in \mathbb{R}^{m},$$

$$\overrightarrow{f} = \begin{bmatrix} f(x^{(1)}) \\ f(x^{(2)}) \\ \vdots \\ f(x^{(m)}) \end{bmatrix} \in \mathbb{R}^{m}, \ \overrightarrow{\epsilon} = \begin{bmatrix} \epsilon^{(1)} \\ \epsilon^{(2)} \\ \vdots \\ \epsilon^{(m)} \end{bmatrix} \in \mathbb{R}^{m},$$

$$X_* = \begin{bmatrix} -(x_*^{(1)})^T- \\ -(x_*^{(2)})^T- \\ \vdots \\ -(x_*^{(m_*)})^T- \end{bmatrix} \in \mathbb{R}^{m_* \times n}, \ \overrightarrow{y_*} = \begin{bmatrix} y_*^{(1)} \\ y_*^{(2)} \\ \vdots \\ y_*^{(m_*)} \end{bmatrix} \in \mathbb{R}^{m_*},$$

$$\overrightarrow{f_*} = \begin{bmatrix} f(x_*^{(1)}) \\ f(x_*^{(2)}) \\ \vdots \\ f(x_*^{(m_*)}) \end{bmatrix} \in \mathbb{R}^{m_*}, \ \overrightarrow{\epsilon_*} = \begin{bmatrix} \epsilon_*^{(1)} \\ \epsilon_*^{(2)} \\ \vdots \\ \epsilon_*^{(m_*)} \end{bmatrix} \in \mathbb{R}^{m_*}.$$

For any $f(\cdot)$ drawn from the GP prior in (2), the marginal distribution

$$\begin{bmatrix} \overrightarrow{f} \\ \overrightarrow{f_*} \end{bmatrix} \Bigg| X, X_* \sim \mathcal{N}\left(\overrightarrow{0}, \begin{bmatrix} K(X,X) & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix}\right) \tag{3}$$

holds where $K(X, X) \in \mathbb{R}^{m \times m}$, $K(X, X_*) \in \mathbb{R}^{m \times m_*}$, $K(X_*, X) \in \mathbb{R}^{m_* \times m}$, and $K(X_*, X_*) \in \mathbb{R}^{m_* \times m_*}$, such that $(K(X, X))_{i,j} = k(x^{(i)}, x^{(j)})$, $(K(X, X_*))_{i,j} = k(x^{(i)}, x_*^{(j)})$, $(K(X_*, X))_{i,j} = k(x_*^{(i)}, x^{(j)})$, $(K(X_*, X_*))_{i,j} = k(x_*^{(i)}, x_*^{(j)})$.

From the i.i.d. noise assumption we have

$$\begin{bmatrix} \overrightarrow{\epsilon} \\ \overrightarrow{\epsilon_*} \end{bmatrix} \sim \mathcal{N}\left(\overrightarrow{0}, \begin{bmatrix} \sigma^2 I & \overrightarrow{0} \\ \overrightarrow{0}^T & \sigma^2 I \end{bmatrix}\right) \tag{4}$$

where I denotes identity matrices of corresponding size. Since the sum of two independent Gaussian random variables is also Gaussian, we get

$$\begin{bmatrix} \overrightarrow{y} \\ \overrightarrow{y_*} \end{bmatrix} \Bigg| X, X_* = \begin{bmatrix} \overrightarrow{f} \\ \overrightarrow{f_*} \end{bmatrix} + \begin{bmatrix} \overrightarrow{\epsilon} \\ \overrightarrow{\epsilon_*} \end{bmatrix} \tag{5}$$

by summing (3) and (4), which yields

$$\begin{bmatrix} \overrightarrow{y} \\ \overrightarrow{y_*} \end{bmatrix} \Bigg| X, X_* \sim \mathcal{N}\left(\overrightarrow{0}, \begin{bmatrix} \Sigma & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) + \sigma^2 I \end{bmatrix}\right). \tag{6}$$

where $\Sigma = K(X, X) + \sigma^2 I$. Now, following the rules for conditioning Gaussians, this yields $y^*|y, X, X^* \sim \mathcal{N}(\mu^*, \Sigma^*)$ where

$$\mu^* = K(X_*, X)\Sigma^{-1}\overrightarrow{y} \tag{7}$$

$$\Sigma^* = K(X_*, X_*) + \sigma^2 I - K(X_*, X)\Sigma^{-1}K(X, X_*) \tag{8}$$

which provides the prediction, or the regression model.

In this paper, we focus explicitly on the squared exponential kernel-based regression. The squared exponential kernel is given by

$$k(x_i, x_j) = \alpha^2 \exp\left(-\frac{1}{2\eta^2}||x_i - x_j||^2\right) \tag{9}$$

When the squared exponential kernel function is used, $\theta \in \mathbb{R}^3$, $\theta = \{\alpha, \eta, \sigma\}$ becomes the set of hyper-parameters. Predictions μ^*, Σ^* are done using an optimized set θ^* obtained by means of minimizing the negative log marginal likelihood [1]; i.e.,

$$\theta^* = \arg\min_{\theta}(-\log[p(\overrightarrow{y}|X, \theta)]) \tag{10}$$

where

$$-\log[p(\overrightarrow{y}|X, \theta)] = \frac{1}{2}X^T\Sigma^{-1}X + \frac{1}{2}\log|\Sigma| + \frac{m}{2}\log(2\pi); \tag{11}$$

$p(\overrightarrow{y}|X, \theta)$ is the conditional probability of $\overrightarrow{y}$ given X, $|\Sigma|$ is the determinant of Σ. Solving (10) is typically done via gradient-based optimization, and this is the point where initializing the hyper-parameter values becomes a concern.

III. Methodology: Initializing Hyper-Parameter Values

A. Pre-Processing Data (Normalizing).

Prior to performing GP regression, normalizing X, X_* and $\overrightarrow{y}$ is proposed. Suppose $\mu_x = [\mu_1, \mu_2, ..., \mu_j, ..., \mu_n]$ is a row vector where μ_j, $j = 1, 2, ..., n$ is the mean of the j^{th} column of X, and $\sigma_x = [\sigma_1, \sigma_2, ..., \sigma_j, ..., \sigma_n]$ is a row vector where σ_j, $j = 1, 2, ..., n$ is the standard deviation (std) of the j^{th} column of X. Now suppose $(X)_{i,j} = x_{(j)}^{(i)}$ and $(X_*)_{i,j} = x_{*(j)}^{(i)}$, and normalization is done for all i and j as follows.

$$x_{(j)}^{(i)} \leftarrow \frac{x_{(j)}^{(i)} - \mu_j}{\sigma_j} \tag{12}$$

$$x_{*(j)}^{(i)} \leftarrow \frac{x_{*(j)}^{(i)} - \mu_j}{\sigma_j} \tag{13}$$

Similarly, suppose μ_y and σ_y are the mean and standard deviation respectively, of the vector $\overrightarrow{y}$. Now, normalization of $\overrightarrow{y}$ is done by performing (14) $\forall\ y^{(i)} \in \overrightarrow{y}$.

$$y^{(i)} \leftarrow \frac{y^{(i)} - \mu_y}{\sigma_y} \tag{14}$$

Note: GP regression is performed on normalized data and from this point onward, X, X_* and $\overrightarrow{y}$ would represent the normalized data.

B. *Hyper-Parameter Initialization.*

The set of hyper-parameter initial values is denoted as $\theta_{ini} = \{\alpha_{ini}, \eta_{ini}, \sigma_{ini}\}$, and the method this paper proposes to determine those initial values is described in this subsection.

To start with, constructing sets $S_1, S_2, ..., S_n$ is proposed where $S_j = \{\overrightarrow{y}, (X)_{:,j}\}$ for $j = 1, 2, ..., n$. $(X)_{:,j}$ is the j^{th} column of X. All S_j, $j = 1, 2, ..., n$ are to be rearranged such that $(X)_{:,j}$ will be sorted to be in ascending order, and $\overrightarrow{y}$ will be sorted correspondingly.

Note: From this point onward, $X_{:,j}$ would represent the sorted j^{th} column of X, and $\overrightarrow{y_j}$ would represent correspondingly sorted $\overrightarrow{y}$, and S_j would represent a sorted set.

Next, constructing the set $d_x = [d_{x1}, d_{x2}, ..., d_{xj}, ..., d_{xn}]$ is proposed where d_{xj} for $j = 1, 2, ..., n$ is given by

$$d_{xj} = \frac{1}{k_{xj}} \sum_{i=1}^{m-1} |(X)_{i,j} - (X)_{i+1,j}| \tag{15}$$

where k_{xj} is the number of instances that $(X)_{i,j} \neq (X)_{i+1,j}$ for $i = 1, 2, ..., m-1$. Then, η_{ini} is set to be the mean of d_x and naturally $\eta_{ini} > 0$ condition would hold.

To determine α_{ini} and σ_{ini}, constructing the set $d_y = [d_{y1}, d_{y2}, ..., d_{yj}, ..., d_{yn}]$ is proposed where d_{yj} for $j = 1, 2, ..., n$ is given by

$$d_{yj} = \frac{1}{k_{yj}} \sum_{i=1}^{m-1} |(\overrightarrow{y_j})^i - (\overrightarrow{y_j})^{i+1}| \tag{16}$$

where $(\overrightarrow{y_j})^i$ is the i^{th} element of $\overrightarrow{y_j}$ and k_{yj} is the number of instances where $(\overrightarrow{y_j})^i \neq (\overrightarrow{y_j})^{i+1}$ for $i = 1, 2, ..., m-1$. Then α_{ini} is set to be the mean of d_y and σ_{ini} is set to be $\sigma_{ini} = \gamma\alpha_{ini} + \delta \min(d_y)$ where $\min(d_y)$ denotes the minimum value of the array d_y and $\gamma, \delta \in \mathbb{R}^+$; $\gamma, \delta < 1$.

There lies a possibility of $\alpha_{ini} = 0$ occurring, and if α in (9) becomes zero, the covariance function values would become zero. Therefore, $\alpha \neq 0$ should hold for the covariance function to produce meaningful values. This imposes that an exception handling routine should come to effect in the rare event of $\alpha_{ini} = 0$ occurring, to indicate that the available training data set is unsuitable to proceed with GP regression, and that more training points are required to be added to make sure $\alpha_{ini} \neq 0$ results. Furthermore, the correct choice of values for γ and δ is an open question, and for the initial investigation done in this paper, the authors set $\gamma = \delta = 0.5$.

IV. Demonstrative Examples and Results

A. *Example 1: One Dimensional Regression Example in (3c) of [26].*

This example (i.e., (3c) of [26]) can be considered a benchmark exercise. Tables I and II show the initial conditions used in [26] and the initial conditions resulting from the technique proposed in this paper, along with the final hyper-parameter values reached. The same training data set (i.e., the 20 training data points shown in Fig. 1) was used with the two initial conditions, and it can be seen from Tables I and II that the same final solution could be reached starting from both initial conditions. The resulting regression model along with the uncertainty and training data are plotted in Fig. 1. The noisy process is given by

$$y^{(i)} = \sin(3x^{(i)}) + \epsilon^{(i)} \tag{17}$$

where $x^{(i)}, y^{(i)}, \epsilon^{(i)} \in \mathbb{R}$; $-0.1373 < \epsilon^{(i)} < 0.1853$, $\forall i$.

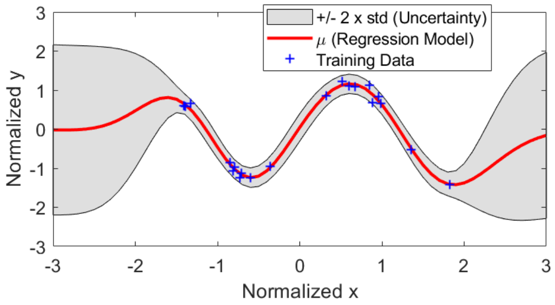

Fig. 1. Final results of Example 1.

B. *Example 2: One Dimensional Regression Example in (4a) of [26].*

In [26], (4a) is solved using a variant of the Matern covariance function. In this paper, we attempted to approximate a regression model for the same data of (4a) using the squared exponential covariance function in conjunction with the proposed hyper-parameter initialization technique. The results shown in Table III and Fig 2 could be achieved.

TABLE I
Hyper-Parameter Values from (3c) of [26], Example 1

Hyper-Parameter	Initial Value	Final Value
α	1	0.5976
η	1	1.0856
σ	0.3679	0.1104

TABLE II
Hyper-Parameter Values from the Proposed Method, Example 1

Hyper-Parameter	Initial Value	Final Value
α	0.4154	0.5976
η	0.1701	1.0856
σ	0.2100	0.1104

TABLE III
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 2

Hyper-Parameter	Initial Value	Final Value
α	0.4170	0.3721
η	0.2027	1.1636
σ	0.2307	0.1177

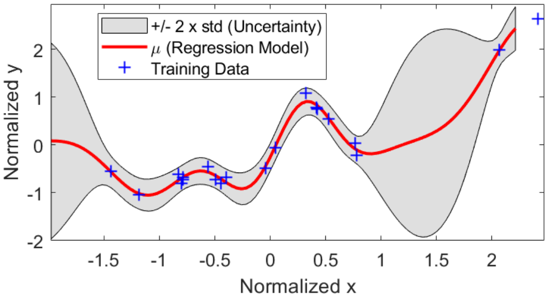

Fig. 2. Final results of Example 2.

TABLE IV
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 3, $\gamma_{noise} = 0.05$ CASE

Hyper-Parameter	Initial Value	Final Value
α	0.2188	0.3800
η	0.0358	1.8160
σ	0.1104	0.1357

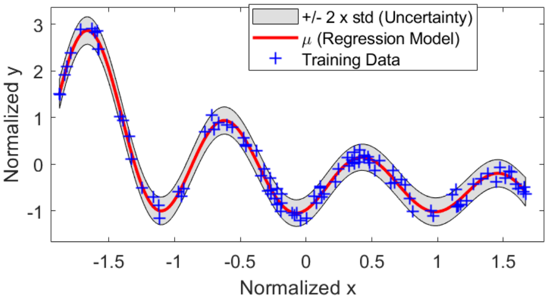

Fig. 3. Final results of Example 3, $\gamma_{noise} = 0.05$ case.

C. Example 3: One Dimensional Regression Example (Synthesized Example).

This subsection presents a synthesized example in finding a regression model for the process shown in (18) with the aim of assessing the impact of noise on the performance of the proposed hyper-parameter initialization technique.

$$y^{(i)} = \exp(-2x^{(i)})\sin(7\pi x^{(i)}) + \exp(-3x^{(i)}) + 0.5 + \epsilon^{(i)} \tag{18}$$

$x^{(i)}, y^{(i)}, \epsilon^{(i)} \in \mathbb{R}, \forall i$. To assess the impact of noise, we define the operators $|*|_e$ and $\max(*)$, where $|*|_e$ replaces each element of an array '$*$' with the element's absolute value, and $\max(*)$ picks the maximum value in an array '$*$' . The criteria in (19) which sets a threshold γ_{noise} for the maximum noise amplitude is then defined. $\gamma_{noise} \in \mathbb{R}^+$, $\gamma_{noise} << 1$.

$$\frac{\max(|\overrightarrow{\epsilon}|_e)}{\max(|\overrightarrow{f}|_e)} \leq \gamma_{noise} \tag{19}$$

The impact of noise is studied by varying the threshold γ_{noise}. Three cases where $\gamma_{noise} = 0.05, 0.15$ and 0.25 are considered where a lower γ_{noise} enforces a lower noise amplitude and a higher γ_{noise} enforces a higher noise amplitude. Tables IV, V and VI show the impact of noise amplitude on hyper-parameter initialization and also the end result. It is evident from the tables that noise does understandably impact the hyper-parameters, however, the resulting regression models and uncertainty margins plotted in Fig. 3, 4 and 5 exhibit that the resulting models are still able to characterize the process behavior.

D. Example 4: One Dimensional Regression Example (A Case With Some Real Data).

In this subsection the initialization technique is experimented on some Pulsed Eddy Current (PEC) sensor data published in [22]. A nature of this data set is that x becomes noisy for coinciding values of y, and when the value of y increases, the variation in x becomes larger. This attribute can be observed in Fig. 6.

The hyper-parameter initializing technique was once again effective as shown by the results in Tables VII and VIII, and Fig. 6. The initial values shown in Table VII are the ones used in [22], and the initial values resulting from the proposed technique are given in Table VIII. It can be seen from those tables that the same final result has been reached starting from both sets of initial values. It can also be seen from Fig. 6 that the estimated hyper-parameters have been able to capture a nonlinear model characterizing the underlying data.

TABLE V
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 3, $\gamma_{noise} = 0.15$ CASE

Hyper-Parameter	Initial Value	Final Value
α	0.4204	0.2684
η	0.0358	1.0623
σ	0.2172	0.3375

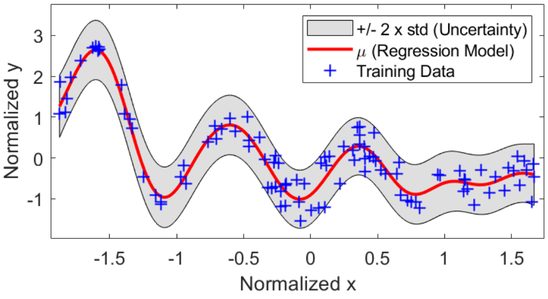

Fig. 4. Final results of Example 3, $\gamma_{noise} = 0.15$ case.

TABLE VI
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 3, $\gamma_{noise} = 0.25$ CASE

Hyper-Parameter	Initial Value	Final Value
α	0.6308	0.2756
η	0.0358	0.9807
σ	0.3252	0.5208

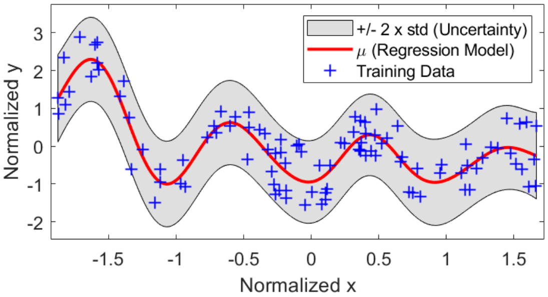

Fig. 5. Final results of Example 3, $\gamma_{noise} = 0.25$ case.

E. Example 5: One Dimensional Regression Example (A Noise-Free Case With Some Real Data).

In this subsection, a noise-free case with minimal data (just 7 data points) is examined using some PEC data published in [27]. Hyper-parameter values along with the regression model that resulted following the proposed initialization technique are shown in Table IX and Fig. 7. Fig. 7 demonstrates how the underlying process has been captured by the regression model.

F. Example 6: A Higher Dimensional Case.

Modeling the surface given by (20) is attempted in this subsection. $x_1^{(i)}, x_2^{(i)}, y^{(i)}, \epsilon^{(i)} \in \mathbb{R}$, $x^{(i)} = \{x_1^{(i)}, x_2^{(i)}\}$, $0 \leq x_1^{(i)}, x_2^{(i)} \leq 1$, $-0.1 \leq \epsilon^{(i)} \leq 0.1$, $\forall i$.

$$y^{(i)} = \exp(-2x_1^{(i)}) \sin(7x_2^{(i)}) + \epsilon^{(i)} \tag{20}$$

Hyper-parameters initialized following the proposed technique were able to tackle this problem as well. Table X, Fig. 8 and Fig. 9 present the results. Fig. 8 shows the regression model surface along with training data; Fig. 9 shows how the uncertainty captures the noise in the data.

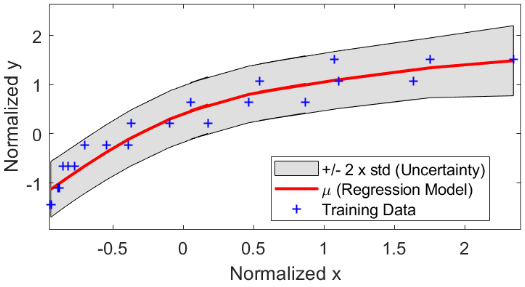

Fig. 6. Final results of Example 4.

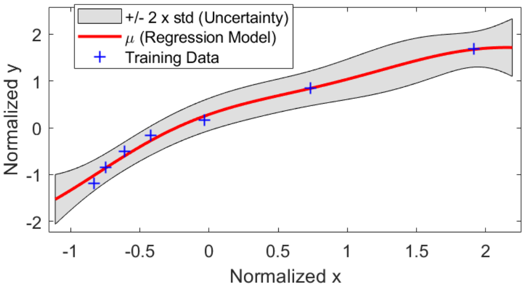

Fig. 7. Final results of Example 5.

V. CONCLUSIONS

A hyper-parameter initialization technique for squared exponential kernel-based GP regression was introduced. The technique includes a data normalization and sorting step, and an absolute difference-based (between adjacent data points) initialization step. In contrast to conventional random initialization and meta cost optimization techniques, the proposed technique brings a degree of novelty by being deterministic; i.e., an exact determination of suitable initial values is possible from training data. Numerous examples going up to higher dimensions were examined, and in all the cases the determined

TABLE VII
HYPER-PARAMETER VALUES FROM [22], EXAMPLE 4

Hyper-Parameter	Initial Value	Final Value
α	7.3891	1.4965
η	2.7183	1.8361
σ	0.1	0.2674

TABLE VIII
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 4

Hyper-Parameter	Initial Value	Final Value
α	0.2417	1.4965
η	0.1428	1.8361
σ	0.1209	0.2674

TABLE IX
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 5

Hyper-Parameter	Initial Value	Final Value
α	0.4789	1.1412
η	0.4574	1.5293
σ	0.4085	0.1342

TABLE X
HYPER-PARAMETER VALUES FROM THE PROPOSED METHOD, EXAMPLE 6

Hyper-Parameter	Initial Value	Final Value
α	0.8023	1.1977
η	0.0175	2.1047
σ	0.4020	0.1641

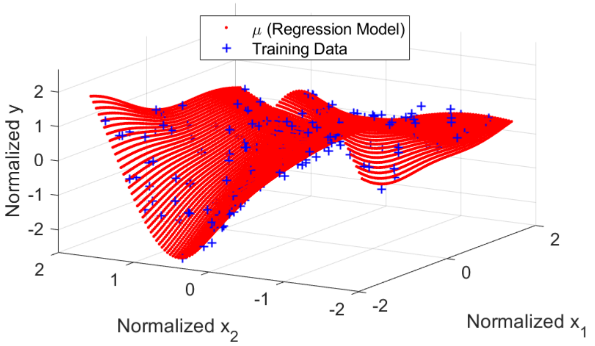

Fig. 8. Regression Model of Example 6 with Training Data.

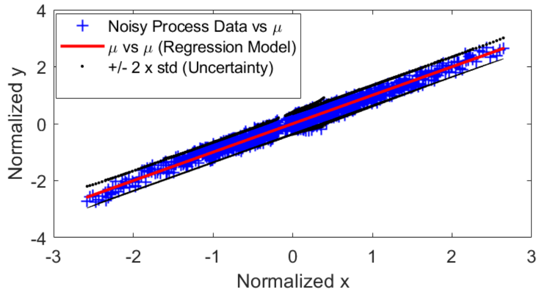

Fig. 9. Uncertainty and Training Data against Regression Model.

initial values were able to lead to final solutions that describe the underlying processes satisfactorily, avoiding the requirement of re-initializing and repeating optimization. Although a guarantee that initializing in the proposed manner would lead to global optimums was not proven in this paper, the results hold evidence for the effectiveness of the regression models produced following this technique. Since this paper studied only the squared exponential kernel-based case, future work can focus on investigating similar initialization techniques for other kernels in relation to GP, and also other machine learning techniques in general. Studying such initialization techniques more deeply will also be interesting to learn whether any failure cases can occur and ways for improvement.

References

[1] C. K. Williams and C. E. Rasmussen, *Gaussian processes for machine learning*. MIT press Cambridge, MA, 2006, vol. 2, no. 3.

[2] ——, "Gaussian processes for regression," in *Advances in neural information processing systems*, 1996, pp. 514–520.

[3] R. B. Gramacy and H. K. H. Lee, "Bayesian treed gaussian process models with an application to computer modeling," *Journal of the American Statistical Association*, vol. 103, no. 483, pp. 1119–1130, 2008.

[4] T. Chen, J. Morris, and E. Martin, "Gaussian process regression for multivariate spectroscopic calibration," *Chemometrics and Intelligent Laboratory Systems*, vol. 87, no. 1, pp. 59–71, 2007.

[5] A. Girard, C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith, "Gaussian process priors with uncertain inputs application to multiple-step ahead time series forecasting," in *Advances in neural information processing systems*, 2003, pp. 545–552.

[6] H. Mori and E. Kurata, "Application of gaussian process to wind speed forecasting for wind power generation," in *2008 IEEE International Conference on Sustainable Energy Technologies*. IEEE, 2008, pp. 956–959.

[7] E. Conte, A. De Maio, and G. Ricci, "Recursive estimation of the covariance matrix of a compound-gaussian process and its application to adaptive cfar detection," *IEEE Transactions on Signal Processing*, vol. 50, no. 8, pp. 1908–1915, 2002.

[8] J. Kocijan, R. Murray-Smith, C. E. Rasmussen, and A. Girard, "Gaussian process model based predictive control," in *Proceedings of the 2004 American control conference*, vol. 3. IEEE, 2004, pp. 2214–2219.

[9] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs," in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[10] J. Hensman, N. Fusi, and N. D. Lawrence, "Gaussian processes for big data," *arXiv preprint arXiv:1309.6835*, 2013.

[11] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, 2011, pp. 2546–2554.

[12] D. Maclaurin, D. Duvenaud, and R. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *International Conference on Machine Learning*, 2015, pp. 2113–2122.

[13] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast bayesian optimization of machine learning hyperparameters on large datasets," *arXiv preprint arXiv:1605.07079*, 2016.

[14] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter, and K. Leyton-Brown, "Auto-weka 2.0: Automatic model selection and hyperparameter optimization in weka," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 826–830, 2017.

[15] M. Feurer, J. T. Springenberg, and F. Hutter, "Initializing bayesian hyperparameter optimization via meta-learning," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[16] M. Wistuba, N. Schilling, and L. Schmidt-Thieme, "Learning hyperparameter optimization initializations," in *2015 IEEE international conference on data science and advanced analytics (DSAA)*. IEEE, 2015, pp. 1–10.

[17] N. Ulapane, A. Alempijevic, T. Vidal-Calleja, J. V. Miro, J. Rudd, and M. Roubal, "Gaussian process for interpreting pulsed eddy current signals for ferromagnetic pipe profiling," in *2014 9th IEEE Conference on Industrial Electronics and Applications*. IEEE, 2014, pp. 1762–1767.

[18] N. N. Ulapane and S. G. Abeyratne, "Gaussian process for learning solar panel maximum power point characteristics as functions of environmental conditions," in *2014 9th IEEE Conference on Industrial Electronics and Applications*. IEEE, 2014, pp. 1756–1761.

[19] A. M. N. N. B. Ulapane, "Nondestructive evaluation of ferromagnetic critical water pipes using pulsed eddy current testing," Ph.D. dissertation, University of Technology Sydney, 2016.

[20] K. Thiyagarajan, S. Kodagoda, and N. Ulapane, "Data-driven machine learning approach for predicting volumetric moisture content of concrete using resistance sensor measurements," in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2016, pp. 1288–1293.

[21] K. Thiyagarajan and S. Kodagoda, "Analytical model and data-driven approach for concrete moisture prediction," in *ISARC 2016-33rd International Symposium on Automation and Robotics in Construction*, 2016.

[22] N. Ulapane, A. Alempijevic, T. Vidal Calleja, and J. Valls Miro, "Pulsed eddy current sensing for critical pipe condition assessment," *Sensors*, vol. 17, no. 10, p. 2208, 2017.

[23] K. Thiyagarajan, "Robust sensor technologies combined with smart predictive analytics for hostile sewer infrastructures," Ph.D. dissertation, 2018.

[24] N. Ulapane, L. Nguyen, J. V. Miro, and G. Dissanayake, "A solution to the inverse pulsed eddy current problem enabling 3d profiling," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, 2018, pp. 1267–1272.

[25] J. Valls Miro, N. Ulapane, L. Shi, D. Hunt, and M. Behrens, "Robotic pipeline wall thickness evaluation for dense nondestructive testing inspection," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1293–1310, 2018.

[26] "Documentation for gpml matlab code version 4.2," http://www.gaussianprocess.org/gpml/code/matlab/doc/, accessed: 2020-01-22.

[27] N. Ulapane, K. Thiyagarajan, D. Hunt, and J. Valls Miro, "Quantifying the relative thickness of conductive ferromagnetic materials using detector coil-based pulsed eddy current sensors," *J. Vis. Exp. (155), e59618*, 2020.